



1ST EDITION

Demystifying Cryptography with OpenSSL 3.0

Discover the best techniques to enhance
your network security with OpenSSL 3.0

A decorative orange geometric shape, resembling a stylized arrow or a series of connected lines, pointing towards the bottom right.

ALEXEI KHLEBNIKOV

Foreword by Jarle Adolfsen, serial entrepreneur, CTO at bspoke, former CTO at Link
Mobility, and a pioneer in computer graphics in the late 1980s and early 1990s

Demystifying Cryptography with OpenSSL 3.0

Discover the best techniques to enhance your network security
with OpenSSL 3.0

Alexei Khlebnikov



BIRMINGHAM—MUMBAI

Demystifying Cryptography with OpenSSL 3.0

Copyright © 2022 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author nor Packt Publishing or its dealers and distributors will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Associate Group Product Manager: Mohd Riyan Khan

Publishing Product Manager: Shrilekha Malpani

Senior Content Development Editor: Adrija Mitra

Technical Editor: Nithik Cheruvakodan

Copy Editor: Safis Editing

Book Project Manager: Kirti Pisat

Proofreader: Safis Editing

Indexer: Sejal Dsilva

Production Designer: Alishon Mendonca

Marketing Coordinator: Ankita Bhonsle

First published: October 2022

Production reference: 1071022

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK.

978-1-80056-034-5

www.packt.com

To my beloved mother, Tatyana Khlebnikova, who, through a lot of effort, care, love, and support, brought me up to the point where I could continue my own development further.

– Alexei Khlebnikov

Foreword

Having been a coder for more than three decades, I've come across fellow techies who could write a book, or even 10, about common but intricate technologies many times. However, very rarely have I come across someone who can explore topics as deeply as Alexei Khlebnikov.

I've had the pleasure of working with him and being his friend for several years now and I am looking forward to really diving into this book. I would certainly say it's a subject that deserves our greatest attention.

In this book, we will be taken on a journey through the basics of OpenSSL, general cryptography, cryptography modes, the “joys” of certificates, and the making of TLS connections. All in great detail if I understand Alexei correctly, which I am very certain I do.

This is an important book, looking closely at technologies we take for granted and that are used basically everywhere to secure our online presence.

There are practical examples and step-by-step explanations of essential concepts to help you along. By the end of the book, you'll be able to use the most popular features of OpenSSL in your products, whether web or desktop.

The book is certainly interesting for the doers, but also for managers and others who think security is important but lack knowledge about it. Don't worry – an in-depth understanding of mathematics is not needed to read this book and learn from it.

In my view as a lifelong techie, learning new things or maybe diving deeper into topics you have some starting knowledge on is rewarding and helpful and it keeps us all on our toes!

Take it away, Alexei!

– Jarle Adolfsen

Serial entrepreneur, CTO at bspoke, former CTO at Link Mobility, and a pioneer in computer graphics in the late 1980s and early 1990s

Contributors

About the author

Alexei Khlebnikov has more than 20 years of professional experience in IT where he has worked in a host of different roles – software developer, system administrator, DevOps engineer, technical leader, architect, and project manager. During these years, Alexei has worked with many technologies – security, artificial intelligence, web development, embedded, mobile, and robotics. Among other companies, Alexei worked for Opera Software on the famous Opera internet browser. Alexei has always been interested in security. He was one of the maintainers of the security-related Opera browser modules, responsible for cryptography, SSL/TLS, and integration with OpenSSL. He was also a member of the security architect group, responsible for the security of the Opera browser. Now, Alexei lives in Oslo, Norway, and works as a senior consultant for bspoke AS. He is also the leader of the architect group at his current employer.

First and foremost, I want to thank my beloved wife, Larisa, and son, Dmitry, for their continued love and support, and for supporting me while I was writing this book and spending less time with them. I also want to thank all the editors, managers, and other people at Packt Publishing who worked on this book, as well as the technical reviewer, Kris. Their help and valuable advice helped me improve the book to the benefit of its readers.

About the reviewer

Krzysztof Kwiatkowski is a cryptography engineer who focuses on problems at the intersection of cryptographic research and implementation. He holds an MSc degree in mathematics with a specialization in computational methods. With a career spanning over 15 years, Kris has worked on a variety of topics related to cryptography, communication, and software security from compact embedded to large distributed systems. Currently, he is concentrating on the implementation of modern, quantum-safe cryptographic schemes, and helping organizations to migrate towards them.

I'd like to thank my wonderful and loving family, who understand my busy schedule and always stand by my side.

Table of Contents

Preface

xv

Part 1: Introduction

1

OpenSSL and Other SSL/TLS Libraries 3

What is OpenSSL?	3	Comparing OpenSSL with	
The history of OpenSSL	4	lightweight TLS libraries	8
What's new in OpenSSL 3.0?	5	Comparing OpenSSL with LibreSSL	9
Comparing OpenSSL with GnuTLS	6	Comparing OpenSSL with BoringSSL	10
Comparing OpenSSL with NSS	7	Summary	11
Comparing OpenSSL with Botan	7		

Part 2: Symmetric Cryptography

2

Symmetric Encryption and Decryption 15

Technical requirements	16	Reviewing DES and 3DES ciphers	21
Understanding symmetric encryption	16	Reviewing the RC4 cipher	22
An overview of the symmetric ciphers supported by OpenSSL	17	Reviewing the ChaCha20 cipher	22
Comparing block ciphers and stream ciphers	17	Reviewing other symmetric ciphers supported by OpenSSL	23
Understanding symmetric cipher security	19	Block cipher modes of operation	25
How many bits of security is enough?	20	Reviewing the Electronic Code Book mode	25
Reviewing the AES cipher	20	Reviewing CBC mode	27

Reviewing CTR mode	28	OpenSSL library	41
Reviewing GCM	30	How to compile and link with OpenSSL	42
Reviewing AES-GCM-SIV	32	How to encrypt with AES programmatically	44
Other block cipher operation modes	32	Implementing the encryption program	45
Choosing the block cipher operation mode	32	Running the encrypt program	49
Padding for block ciphers	33	How to decrypt with AES programmatically	49
How to generate a symmetric encryption key	34	Implementing the decrypt program	50
Downloading and installing OpenSSL	35	Running the decrypt program	52
How to encrypt and decrypt with AES on the command line	37	Summary	53
Initializing and uninitializing			

3

Message Digests 55

Technical requirements	55	Reviewing the SHA-3 family of hash functions	61
What are message digests and cryptographic hash functions?	56	Reviewing the SHA-1 and SHA-0 hash functions	63
Why are message digests needed?	56	Reviewing the MD family of hash functions	64
Data integrity verification	56	Reviewing the BLAKE2 family of hash functions	64
Basis for HMAC	57	Reviewing less popular hash functions supported by OpenSSL	65
Digital signatures	57	Which cryptographic hash function should you choose?	66
Network protocols	57	How to calculate a message digest on the command line	67
Password verification	58	How to calculate the message digest programmatically	68
Content identifier	58	Implementing the digest program	68
Blockchain and cryptocurrencies	58	Running the digest program	69
Proof-of-work	58	Summary	70
Assessing the security of cryptographic hash functions	59		
Overview of the cryptographic hash functions supported by OpenSSL	60		
Reviewing the SHA-2 family of hash functions	60		

4

MAC and HMAC 71

Technical requirements	71	How to calculate HMAC on the command line	77
What is a MAC?	72	How to calculate HMAC programmatically	78
Understanding MAC function security	72	Implementing the hmac program	79
HMAC – a hash-based MAC	73	Running the hmac program	81
MAC, encryption, and the Cryptographic Doom Principle	75	Summary	81

5

Derivation of an Encryption Key from a Password 83

Technical requirements	83	Deriving a key from a password on the command line	87
Understanding the differences between a password and an encryption key	84	Deriving a key from a password programmatically	88
What is a key derivation function?	84	Implementing the kdf program	89
Overview of key derivation functions supported by OpenSSL	87	Running the kdf program	90
		Summary	91

Part 3: Asymmetric Cryptography and Certificates

6

Asymmetric Encryption and Decryption 95

Technical requirements	95	Verifying a key fingerprint over the phone	97
Understanding asymmetric encryption	96	Key splitting	98
Understanding a Man in the Middle attack	96	Signing the key by a trusted third party	98
Meeting in person	97	What kind of asymmetric encryption is available in OpenSSL?	98

Understanding a session key	99	Running the rsa-encrypt program	110
Understanding RSA security	99	Understanding the OpenSSL error queue	111
How to generate an RSA keypair	102	How to decrypt with RSA programmatically	115
How to encrypt and decrypt with RSA on the command line	105	Implementing the rsa-decrypt program	116
How to encrypt with RSA programmatically	107	Running the rsa-decrypt program	117
Implementing the rsa-encrypt program	108	Summary	117

7

Digital Signatures and Their Verification 119

Technical requirements	120	How to generate an elliptic curve keypair	126
Understanding digital signatures	120	How to sign and verify a signature on the command line	127
Difference between digital signatures and MACs	121	How to sign programmatically	129
Overview of digital signature algorithms supported by OpenSSL	121	Implementing the ec-sign program	130
Reviewing RSA	121	Running the ec-sign program	132
Reviewing DSA	122	How to verify a signature programmatically	133
Reviewing ECDSA	123	Implementing the ec-verify program	134
Reviewing EdDSA	124	Running the ec-verify program	135
Reviewing SM2	125	Summary	136
Which digital signature algorithm should you choose?	125		

8

X.509 Certificates and PKI 137

Technical requirements	137	What are X509v3 extensions?	146
What is an X.509 certificate?	138	Understanding X.509 Public Key Infrastructure	147
Understanding certificate signing chains	140	How to generate a self-signed certificate	147
How are X.509 certificates issued?	144		

How to generate a non-self-signed certificate	150	programmatically	154
How to verify a certificate on the command line	154	Implementing the x509-verify program	155
How to verify a certificate		Running the x509-verify program	158
		Summary	159

Part 4: TLS Connections and Secure Communication

9

Establishing TLS Connections and Sending Data over Them 163

Technical requirements	164	Understanding OpenSSL BIOs	174
Understanding the TLS protocol	164	Establishing a TLS client connection programmatically	176
Understanding a TLS handshake	166	Implementing the tls-client program	177
What happens after the TLS handshake?	167	Running the tls-client program	182
The history of the TLS protocol	168	Accepting a TLS server connection programmatically	183
Establishing a TLS client connection on the command line	170	Implementing the tls-server program	184
Preparing certificates for a TLS server connection	172	Running the tls-server program	190
Accepting a TLS server connection on the command line	173	Summary	192

10

Using X.509 Certificates in TLS 193

Technical requirements	193	Using Certificate Revocation Lists in C programs	202
Custom verification of peer certificates in C programs	194	Registering the CRL lookup callback	205
Registering the verification callback	197	Implementing the CRL lookup callback	206
Implementing the verification callback	198	Implementing the function for downloading a CRL from a distribution point	208
Running the program	200		

Implementing the function for downloading a CRL from an HTTP URL	210	Using TLS client certificates	225
Running the program	211	Generating TLS client certificates	225
Using the Online Certificate Status Protocol	213	Packaging client certificates into PKCS #12 container files	226
Understanding the Online Certificate Status Protocol	213	Requesting and verifying a TLS client certificate on the server side programmatically	228
Using OCSP on the command line	214	Establishing a TLS client connection with a client certificate programmatically	234
Using OCSP in C programs	218	Summary	241

11

Special Usages of TLS 243

Technical requirements	243	Using TLS on non-blocking sockets	254
Understanding TLS certificate pinning	244	Changing the run_tls_client() function	254
Using TLS certificate pinning	246	Running the tls-client-non-blocking program	260
Changing the run_tls_client() function	247	Understanding TLS on non-standard sockets	262
Implementing the cert_verify_callback() function	249	Using TLS on non-standard sockets	263
Running the tls-cert-pinning program	251	Implementing the service_bios() function	264
Understanding blocking and non-blocking sockets	253	Reimplementing the run_tls_client() function	266
		Running the tls-client-memory-bio program	273
		Summary	275

Part 5: Running a Mini-CA

12

Running a Mini-CA 279

Technical requirements	279	Generating an intermediate CA certificate	287
Understanding the openssl ca subcommand	280	Generating a certificate for a web server	292
Generating a root CA certificate	281		

Generating a certificate for a web and email client	294	Providing certificate revocation status via OCSP	301
Revoking certificates and generating CRLs	296	Summary	304
Index			305

Other Books You May Enjoy	318
----------------------------------	------------

Preface

Security and networking are essential features of software today. The modern internet is full of worms, Trojan horses, men-in-the-middle, and other threats. This is why maintaining security is more important than ever.

OpenSSL is one of the most widely used and essential open-source projects on the internet for this purpose. If you are a software developer, system administrator, network security engineer, or DevOps specialist, you've probably stumbled upon this toolset in the past – but how do you make the most out of it? With the help of this book, you will learn the most important features of OpenSSL, and gain insight into its full potential.

This book contains step-by-step explanations of essential cryptography and network security concepts, as well as practical examples illustrating usage of those concepts. You'll start by learning the basics such as how to perform symmetric encryption and calculate message digests. Next, you will discover more about cryptography: MAC and HMAC, public and private keys, and digital signatures. As you progress, you will explore best practices for using X.509 certificates, public key infrastructure, and TLS connections.

By the end of this book, you'll be able to use the most popular features of OpenSSL, allowing you to implement cryptography and TLS in your applications and network infrastructure.

Who this book is for

This book is for software developers, system administrators, DevOps specialists, network security engineers, and analysts, or anyone who wants to keep their applications and infrastructure secure. Software developers will learn how to use the OpenSSL library to empower their software with cryptography and TLS. DevOps professionals and sysadmins will learn how to work with cryptographic keys and certificates on the command line, and how to set up a mini-CA for their organization. A basic understanding of security and networking is required.

What this book covers

Chapter 1, OpenSSL and Other SSL/TLS Libraries, will outline what OpenSSL is and what its strengths are and take a look into OpenSSL's history and at what's new in OpenSSL 3.0. We will also compare OpenSSL to other SSL/TLS libraries.

Chapter 2, Symmetric Encryption and Decryption, will cover the important concepts in symmetric encryption – ciphers, encryption modes, and padding. We will overview modern ciphers, encryption modes, and padding types and recommend which technology to use in which situation. Usage of these technologies will be illustrated by command-line and C code examples.

Chapter 3, Message Digests, will explore why message digests, also known as cryptographic hashes, are needed and where they are used. We will get an overview of modern cryptographic hash functions that calculate message digests and recommend which hash function to use in which situation. The calculation of message digests will be illustrated by command-line and C code examples.

Chapter 4, MAC and HMAC, will explain why **Message Authentication Codes (MACs)** are needed and where they are used. Since it's a popular MAC type, **Hash-based MAC (HMAC)** will be discussed. We will also learn about how to combine HMAC with encryption and about the Cryptographic Doom Principle. The calculation of HMAC will be illustrated by a code example.

Chapter 5, Derivation of an Encryption Key from a Password, will show why a password itself cannot be used for encryption and why key derivation is needed. We will overview modern key derivation functions and recommend which one to use when. Then, encryption key derivation will be illustrated by command-line and C code examples.

Chapter 6, Asymmetric Encryption and Decryption, will unpack why asymmetric encryption is needed, how it works, and how private and public keys are used to achieve encryption and decryption. Encryption and decryption using RSA will be illustrated by command-line and C code examples.

Chapter 7, Digital Signatures and Their Verification, will clarify why digital signatures are needed and where they are used. We will overview modern digital signature algorithms, such as RSA, ECDSA, and EdDSA, and recommend which digital signature scheme to use in which situation. Digital signing and signature verification will be illustrated by command-line and C code examples.

Chapter 8, X.509 Certificates and PKI, will detail what X.509 certificates are, why they are needed, and where they are used. We will also explain how certificates sign other certificates and how certificate signing chains are formed, as well as what **Public Key Infrastructure (PKI)** is and how certificate verification is used to verify identities – for example, the identities of websites. The usage of the techniques mentioned will be illustrated by command-line and C code examples.

Chapter 9, Establishing TLS Connections and Sending Data over Them, will break down what the TLS protocol is, why it is needed, and why it is used so widely. We will also learn what the difference between SSL and TLS is. Then, we will learn how to establish and shut down a TLS connection, as well as how to send and receive data over TLS. Working with TLS will be illustrated by command-line and C code examples.

Chapter 10, Using X.509 Certificates in TLS, will elaborate on how to work with X.509 certificates in TLS and why certificates are important for TLS. We will also learn how to verify a remote certificate. Then, we will learn how to further check the certificate validity using a CRL and OCSP. Finally, we will learn how to use a client certificate. Working with certificates will be illustrated by command-line and C code examples.

Chapter 11, Special Usages of TLS, will look into special usages of TLS: TLS pinning, using non-blocking networking mode, and TLS connections over non-standard sockets or special networking layers using OpenSSL **Basic Input-Output Objects (BIOs)**. The usage of the techniques mentioned will be illustrated by C code examples.

Chapter 12, Running a Mini-CA, will instruct you on how to run your own mini-CA in order to control certificates and build PKI into an organization. Running a mini-CA will be illustrated by example configuration files and commands.

To get the most out of this book

You will have to install OpenSSL on your computer in order to run the command-line and C code examples. If you haven't installed it yet, *Chapter 2, Symmetric Encryption and Decryption*, will help you to do so. To build the C code examples, you will need a C11-compatible C compiler and a linker. You will have to install these development tools following their respective documentation. All the examples have been tested on Kubuntu Linux 22.04 using GNU C Compiler, GNU Linker (LD), and GNU Make from the Linux distribution mentioned. Other development tools, such as LLVM Clang or Microsoft Visual C++, should also be compatible with the code examples in this book.

Software/hardware covered in the book	System requirements
OpenSSL 3.0	<ul style="list-style-type: none">• Linux, FreeBSD, macOS, Windows, or any other OS supported by OpenSSL• A C compiler – for example, GNU C Compiler• A linker – for example, GNU Linker (LD)• Your favorite C/C++ IDE or code editor• A build tool – for example, GNU Make (optional)

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to copying and pasting code.

While explanations of OpenSSL features and code examples are sometimes very detailed, the book is meant to provide guidance, not to replace the OpenSSL documentation. If you are wondering about details of OpenSSL functionality that are not covered by the book, feel free to consult the OpenSSL documentation, the OpenSSL source code, or just experiment with your own code using OpenSSL!

Download the example code files

You can download the example code files for this book from GitHub at <https://github.com/PacktPublishing/Demystifying-Cryptography-with-OpenSSL-3>. If there's an update to the code, it will be updated in the GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing/>. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: <https://packt.link/cOWEO>.

Conventions used

There are a number of text conventions used throughout this book.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: “SSH user public keys are pinned on the server in the `authorized_keys` file.”

A block of code is set as follows:

```
if (pinned_server_cert)
    X509_free(pinned_server_cert);
if (pinned_server_cert_file)
    fclose(pinned_server_cert_file);
```

Any command-line input or output is written as follows:

```
$ ./tls-server 4433 server_keypair.pem server_cert.pem
*** Listening on port 4433
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: “Reduced maintenance because you don’t need to make a **Certificate Signing Request (CSR)** and communicate with a CA. You can even use a self-signed certificate.”

Tips or Important Notes

Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at customercare@packtpub.com and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit www.packtpub.com/support/errata and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit authors.packtpub.com.

Share Your Thoughts

Once you've read *Demystifying Cryptography with OpenSSL 3*, we'd love to hear your thoughts! Please [click here](#) to go straight to the Amazon review page for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

